Computing and decomposing tensors with Tensorlab: Exercises

Paul Breiding and Nick Vannieuwenhoven

Download Tensorlab at `https://www.tensorlab.net`. Download the Matlab code for computing the geometric condition number and the pencil-based algorithm at `http://personal-homepages.mis.mpg.de/breiding/tensors_cond_and_pba.zip`. Uncompress the archive and add the files to the current Matlab session by using `addpath`.

Recall that you can use `help x` or `doc x` to learn more about `x` in Matlab.

Exercise 1

1. Generate a random real $3 \times 3 \times 2$ tensor `T` of rank 3 by sampling the elements of the factor matrices $A \in \mathbb{R}^{3\times 3}, B \in \mathbb{R}^{3\times 3}, C \in \mathbb{R}^{2\times 3}$ independently from a standard normal distribution. In Tensorlab the tensor with factor matrices A, B, C is generated with `T = cpdgen({A,B,C})`. Compute the rank-3 CPD of `T` by using `cpd(T,3)`. What is the output of `cpd`? Compare the tensor represented by the output to `T` by computing the Frobenius norm of their difference using `frob`.
2. Implement a function to compute the Kruskal rank of an $m \times n$ matrix. How do you numerically compute the rank of a matrix? Is your strategy trustworthy?
3. Compute the Kruskal ranks of the factor matrices. Is the decomposition unique?

Exercise 2

1. Implement an algorithm for computing the multilinear multiplication

$$(M_1, M_2, M_3) \cdot \mathcal{T}$$

   for $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $M_k \in \mathbb{R}^{m_k \times n_k}$. Tensorlab's `tens2mat` and `mat2tens` functions can be helpful. What is are the dimensions of the output tensor?
2. Implement an algorithm for computing the higher-order singular value decomposition (HOSVD), using the `svd` function in Matlab; use the compact SVD (called "economy size" by Matlab) where possible. The output should be the core tensor, the basis matrices, and the multilinear rank as output. You can test the correctness of your implementation by comparing your output to Tensorlab's `mlsvd` function. What is the multilinear rank of a random real $16 \times 8 \times 18$ tensor? What is the multilinear rank of a random real $16 \times 8 \times 18$ tensor of rank 3? Why?

Exercise 3

Create a $50 \times 50 \times 2500$ tensor $\mathcal{T}$ of multilinear rank $(2,2,2)$. Compute the standard T-HOSVD and the ST-HOSVD of $\mathcal{T}$ and compare the speed of the computations. In this context, the Matlab functions `tic` and `toc` may be useful, as well as Tensorlab's `mlsvd` function. What are the computational complexities of T-HOSVD and ST-HOSVD when applied to $n \times n \times n^2$ tensors? Recall that computing the SVD of an $m \times n$ matrix has complexity $\mathcal{O}(\min\{mn^2, nm^2\})$.

Exercise 4

1. Write a function that evaluates the forward error

$$\min_{\pi \in \mathfrak{S}_r} \|A \odot B \odot C - (A' \odot B' \odot C')P_\pi\|_F,$$

   where $\mathfrak{S}_r$ is the group of permutations on r elements and P_π is the permutation matrix (whose columns are a permutation of the columns of an $r \times r$ identity matrix) representing π, between two sets of rank-1 tensors, given by their factor matrices (A, B, C) and (A', B', C').

2. Generate the $4 \times 3 \times 3$ tensor $\mathcal{T}$ with factor matrices

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & -1 & 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

   This tensor has rank 3. Compute a rank-2 approximation of $\mathcal{T}$ in Tensorlab using `cpd(T,2)`. Investigate the output. What are the norms of the rank-1 summands? What is the norm of the tensor? Do you notice anything curious about the factor matrices? Also compute the condition number of the output using `cpdgeomcond`. Can you conclude that the computed rank-1 tensors are close to the rank-1 tensors in a CPD of $\mathcal{T}$?

3. Generate the $4 \times 3 \times 3$ tensor $\mathcal{T}_k$ with factor matrices

$$A_k = \begin{bmatrix} 1 & 0 & 2^{-k} \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}, B_k = \begin{bmatrix} 2^{-k} & 0 & 0 \\ 1 & 1 & 1 \\ 1 & -1 & 1 \end{bmatrix}, C_k = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 2^{-k} & 0 & 0 \end{bmatrix}.$$

   Consider $k \in \{1, 2, 4, 8, 16\}$. Check numerically if $\mathcal{T}_k$ is 3-identifiable according to Kruskal's lemma. Compute the factor matrices (A_k', B_k', C_k') of this unique rank-3 CPD of $\mathcal{T}$ in Tensorlab using `cpd`. How large is the Euclidean distance between $\mathcal{T}_k$ and the tensor represented by the factor matrices (A_k', B_k', C_k')? How large is the forward error between the rank-1 tensors? Can you explain this with the condition number of the CPD using `cpdgeomcond`?

Exercise 5

Generate 20 random real 2×2 matrices whose elements are i.i.d. standard normally distributed. Compute their ranks. Generate 100 random real $2 \times 2 \times 2$ tensors by sampling their elements i.i.d. from a standard normal distribution. How would you compute their ranks? Compute their ranks. Is anything out of the ordinary? What is the empirical distribution? Can you guess the probability that the rank is 2?

Exercise 6

Create a (random) orthogonal matrix $Q \in \mathbb{R}^{5\times 2}$. For instance, this can be done by sampling $A \in \mathbb{R}^{5\times 2}$ with i.i.d. standard normal entries and passing it to Matlab's `orth` function. Create a random tensor $\mathtt{T} \in \mathbb{R}^{7\times 6\times 5}$ of rank 3 (as in exercise 1). Compute the CPD of `T` using the so-called pencil-based algorithm `cpd_pba` using the matrix `Q` for the projection: `cpd_pba(T, 3, Q)`.

Now, let $U \in \mathbb{R}^{5\times 5}$ be an orthogonal matrix whose last two columns are Q. Sample orthogonal matrices $A \in \mathbb{R}^{7\times 3}, B \in \mathbb{R}^{6\times 3}$ and put

$$C_k := 0.4 \cdot U \begin{bmatrix} 1.5 & 1 & 1 \\ -1 & -1.5 & 1 \\ -1 & 1 & -1.5 \\ -1 & 1 & 1 \\ -1 + 10^{-k} & 1 & 1 \end{bmatrix}.$$

Let $\mathtt{S}_k$ be the tensor with factor matrices A, B, C_k. Compute the condition number of this CPD of $\mathtt{S}_k$ using `cpdgeomcond`. Do you expect problems when computing the CPD?

Run `cpd_pba(`$\mathtt{S}_k$`, 3, Q)` for $k \in \{6, 8, 10, 12\}$ and compare the result with the output of `cpd`. Is there something strange? Which algorithm is more accurate?

Finally, compute rank-3 CPDs of `T`, $(I_7, I_6, Q^T)\cdot\mathtt{T}$, $\mathtt{S}_k$ and $(I_7, I_6, Q^T)\cdot\mathtt{S}_k$, where I_n is the $n \times n$ identity matrix. Compare the condition numbers of the decompositions you get using `cpdgeomcond`. Do you recognize any pattern? Based on the implementation of `cpd_pba` can you conjecture what goes wrong? What is most likely the cause for the observed problems: the mathematical decomposition problem, the algorithm, Matlab, or the computer?

Exercise 7

Get the data set from `http://www.models.life.ku.dk/nwaydata1`. This data set is a $5 \times 51 \times 201$ tensor `T`. The tensor `T` contains data from 5 experiments: three amino acids tryptophan (Trp), tyrosine (Tyr), phenylalanine (Phe) were dissolved in phosphate buffered water (with different amounts in each of the 5 experiments). The resulting substance was measured by fluorescence (excitation 250-300 nm and emission 250-450 nm, 1 nm intervals) on a spectrofluorometer.

1. Visualize the tensor by running `voxel3(T)`.
2. What do you expect to be the rank of `T`? Verify your assertion by first computing an ST-HOSVD compression to rank $(\min\{5, 2r\}, 2r, 2r)$ and then approximating the core tensor by a rank-r CPD of using `cpd`. What is a suitable r?

3. Perform the same experiment but this time by computing a rank-r approximation directly from the large tensor `T`. Do this by creating an options structure `opts = []; opts.Compression = false;` and then compute `cpd(T, r, opts)`, where r is the desired rank. Do you get the same results? Which technique is faster?
4. Have a look at output factor matrices from your computation. Compute the Kruskal ranks. Is the decomposition unique?
5. Compute the condition number of your decomposition.
6. Do you trust the result of the computation? Why or why not?
7. How should the rank-1 summands be interpreted?
8. Prove that the computations in step 1 and 2 are equivalent if `T` has a rank-r CPD.